UNIVERSITY OF BUEA

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

P.O. Box 63,
Buea, South West Region
CAMEROON
Tel : (237) 3332 21 34/3332 26 90
Fax: (237) 3332 22 72

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

COURSE: CEF 440 - Internet Programming and Mobile Programming

TASK 4: System Modeling And Design of the biometric student's attendance app

Presented by:

NAMES	MATRICULES
KENNE DATEWO SUZY MAIVA	FE21A214
SIAHA TOUKO AUBIN	FE21A304
DJEUNOU DJEUNOU MARIEKE JETTIE	FE21A168
TSAPZE ZAMBOU ROSELINE CYNTHIA	FE21A328
MATHO SONKWA HESTIE MAYELLE	FE21A438

Academic year: 2023-2024

COURSE INSTRUCTOR: Dr. NKEMENI VALERY

Table of Contents

INTRODUCTION

As educational institutions strive to improve student engagement and attendance tracking, the development of a biometric student attendance mobile application has become a crucial necessity. This report delves into the system modeling and design of such an application called AttendEase that leverage the power of Unified Modeling Language (UML) to provide a comprehensive understanding of the system's architecture, functionalities, and interactions. By employing various UML diagrams, this report aims to document the key design decisions, workflow processes, and technical specifications that underpin the development of a robust and user-friendly biometric attendance solution for educational institutions.

UML is a way to use diagrams to visualize a system or database. The objective of UML is to provide a standardized notation that can be used by all objects of object-oriented methods. There are two types of UML diagram, which are structural diagrams, and Behavioral diagrams. These categories have many instances of UML diagrams that can be shown in the *figure 1* below, but for the AttendEase biometric mobile application, we will make use of sixth of them: the context diagram, the use case diagram, the class diagram, the sequence diagram, the deployment diagram, and the activity diagrams.

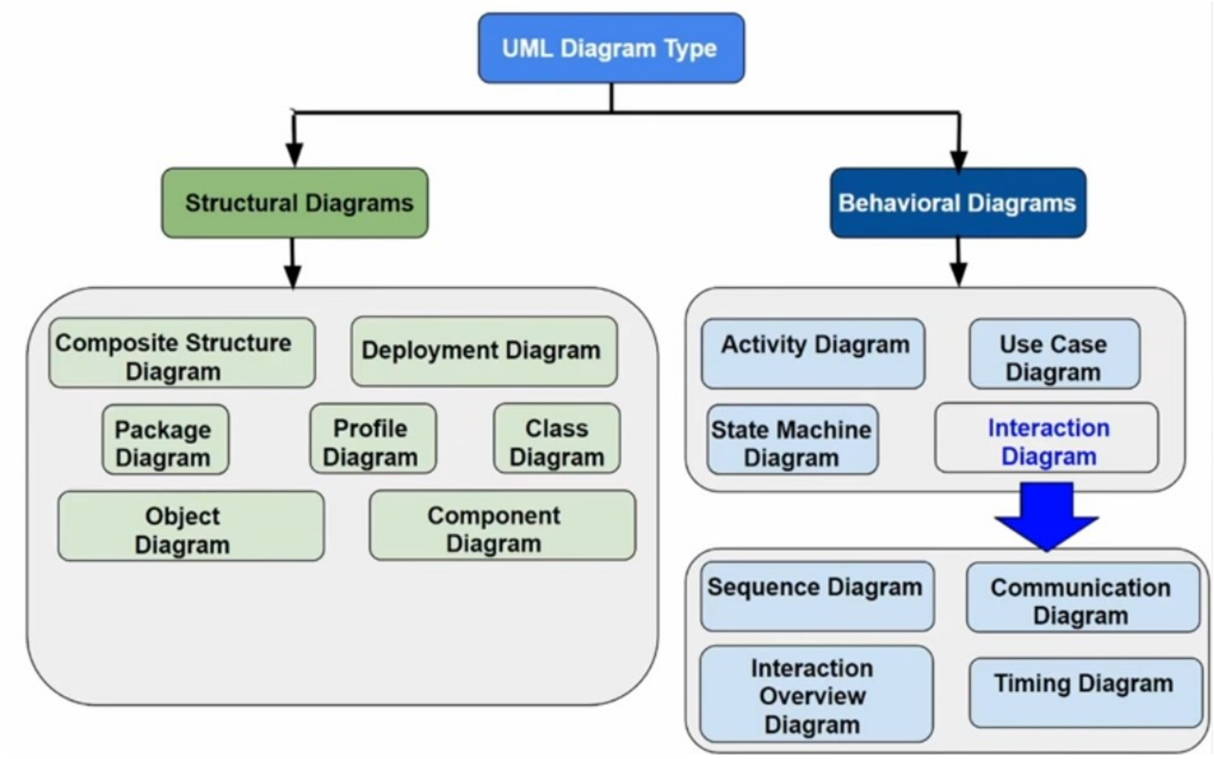


Figure 1: types of UML diagrams based on different categories

1. CONTEXT DIAGRAM

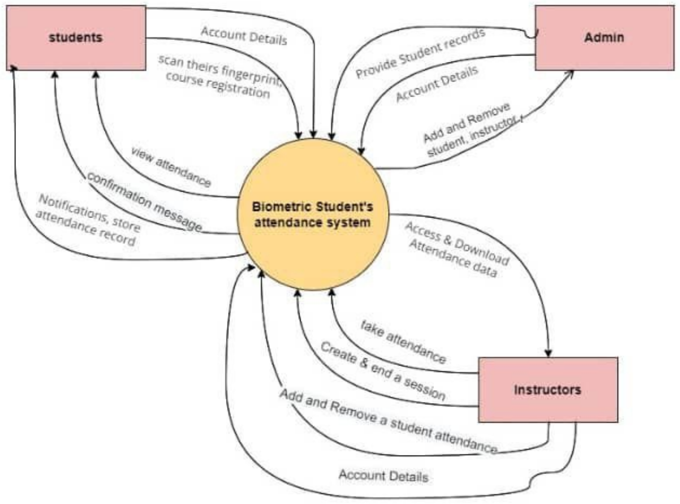
The context diagram is a crucial tool in system modeling as it provides a high-level overview of the proposed AttendEase biometric student attendance mobile application and its interactions with the external entities or stakeholders of our system. This diagram serves as the foundation for understanding the system's boundaries, the key actors involved, and the primary information flows between the application and its surrounding environment. By clearly defining the system's boundaries and identifying the critical interactions, the context diagram lays the groundwork for further exploration and refinement of the application's functionalities, data flows, and integration points, guiding the subsequent modeling efforts and serving as a reference point for the more detailed system design and implementation. In *figure 2* bellow, we can see the context diagram of our mobile application:

🟦 Represent the external entity : admin, student, instructors.

🔵 Represent the main system: biometric student’s attendance system.

→ Represent the flow of data

eFig
ure
2:


students
Admin
Instructors
Biometric Student's attendance system
Account Details
scan theirs fingerprint, course registration
view attendance
confirmation message
Notifications, store attendance record
Provide Student records
Account Details
Add and Remove student, instructor
Access & Download Attendance data
take attendance
Create & end a session
Add and Remove a student attendance
Account Details

Context diagram of AttendEase mobile application

2. USE CASE DIAGRAM

This diagram summarizes some of the relationships between use cases actors and systems, but they don’t show the order in which steps to achieve the goals of each use cases that are taking. It contains a few figures and if it contains more than 20 figures, you are probably out of the box (using the diagram incorrectly). The use case diagram describes the functional requirements of the system in terms of use cases, and they allow us to link what we need to the system with how the system meets these needs. We can clearly see in our system that actors trigger use cases, and the purposes of the use case diagram are:

- Specify the context of the system
- Capture the requirements of the system
- Validate system architecture
- Drive the implementation and generate test cases

In *figure 3* bellow, we can see each use case of our system, the actors, and the cases they are interacting with.

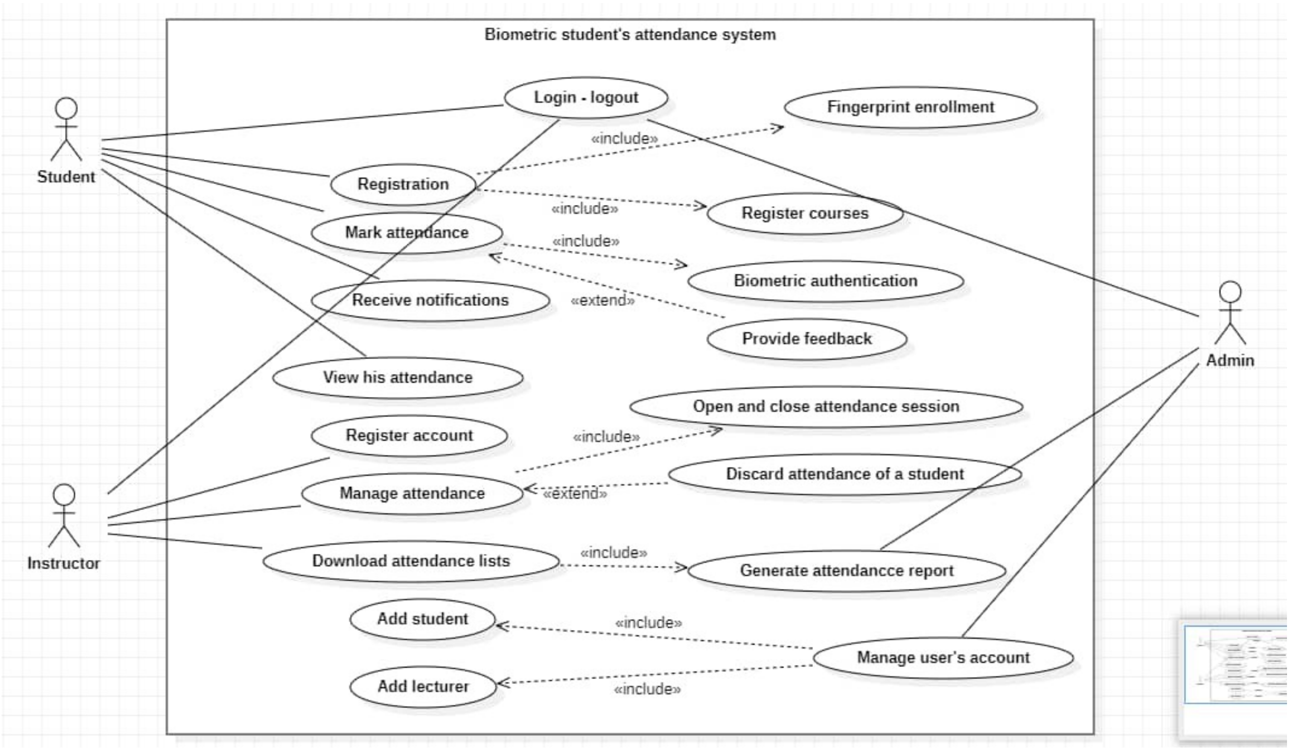


Figure 3: Use case diagram of the system

3. CLASS DIAGRAM

This diagram provides a comprehensive representation of the application's core classes, illustrating the structure of the system by describing classes, their attributes, operations, and the relationships between them. The class diagram focuses on modeling the key entities such as the Student, Instructor, and Attendance classes, and their interconnections, which demonstrate the interdependencies and data flows within the application. By modeling the application's class structure and their interactions in *figure 4* bellow, the class diagram lays the foundation for the system's implementation, ensuring that the data and functionality are organized in a cohesive and efficient manner, and serving as a valuable reference point for developers, architects, and stakeholders throughout the development lifecycle of our mobile application. Here, we have the shape initial state – attributes – behaviors of the system.

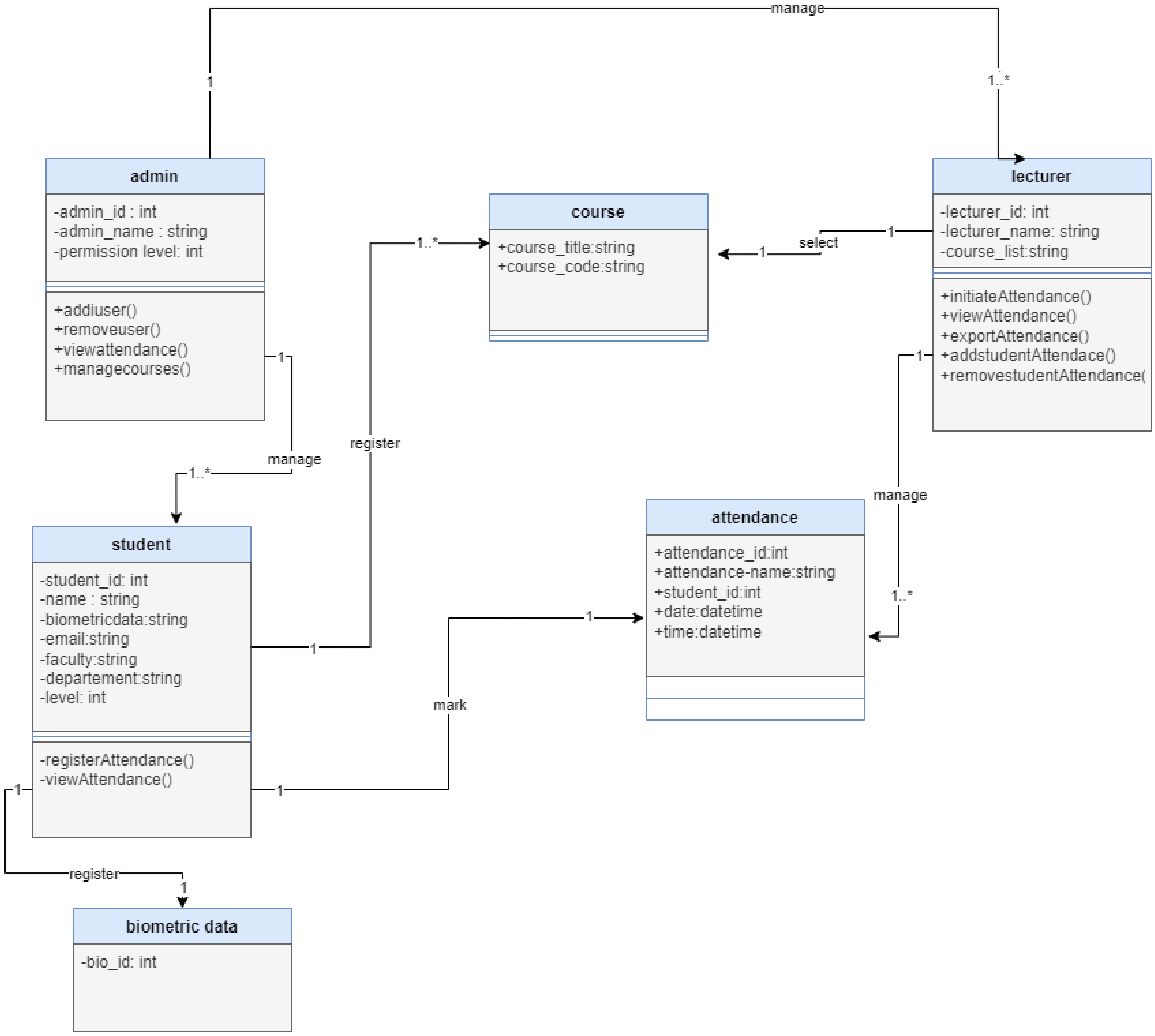


Figure 4: Class Diagram of the system

4. SEQUENCE DIAGRAM

The sequence diagram is a powerful UML tool that helps visualize the dynamic interactions and flow of messages between the various objects, components, and actors within our biometric student attendance mobile application on a time sequence. It shows how objects interact with each other with a particular use case. These are interaction diagram that detail how operations are performed. They capture the interaction between objects in the concepts of collaboration and show how elements interacts over time and organized with respective objects horizontally and time vertically. The horizontal axis shows us elements that participates in the interaction, and they can appear in any order.

In our system, the sequence diagram shown in *figure 5* bellow focuses on illustrating the chronological order of events that occur when a student attempts to mark their attendance using the mobile app, depicting the distinct steps involved, such as the lecturer initiating the attendance process by starting an attendance session, the student login into their account and having the possibilities to view their attendance, the app capturing the student's biometric data, the data being verified against the central database or sent to the backend, and the attendance record being updated accordingly. By modeling these sequential interactions, the sequence diagram elucidates the underlying logic and communication pathways between the student, the mobile app, the authentication service, and the attendance management system, providing valuable insights into the runtime behavior of the application and ensuring that the design aligns with the functional requirements of the biometric attendance solution.

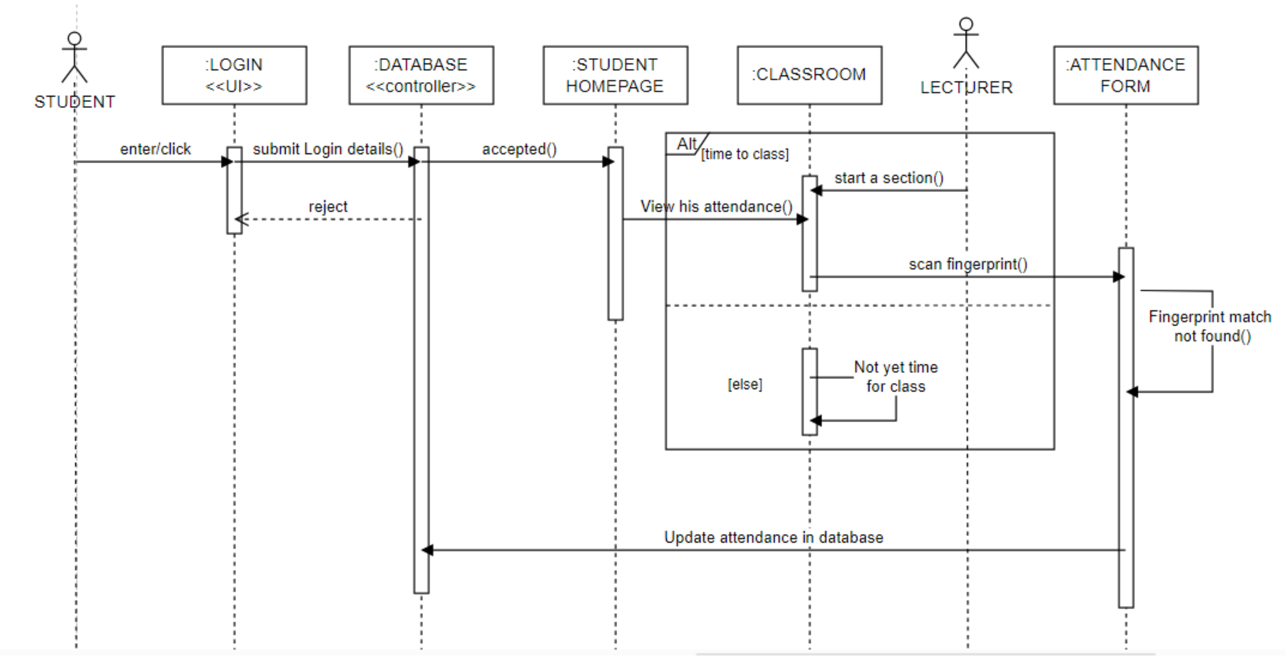


Figure 5: Sequence Diagram of AttendEase, mobile application

LOGIN<<UI>>: Represents the login user interface where the student can log in to the system.

DATABASE<<controller>>: Represents the database controller that handles user authentication and other data-related operations.

STUDENT HOMEPAGE: Represents the student's homepage, where they can access various functionalities.

CLASSROOM: Represents the classroom where the attendance session is managed.

ATTENDANCE FORM: Represents the attendance form, where the student can mark their attendance.

LECTURER: Represents the lecturer, who starts/end the attendance session

5. DEPLOYMENT DIAGRAM

This diagram helps to model the physical aspect of an object-oriented software system. It is mainly focused on two parts: the client application and the server application. The *figure 6* below can show us the main architecture of any deployment diagram, and in *figure 7*, we can see the deployment diagram of the AttendEase mobile application:

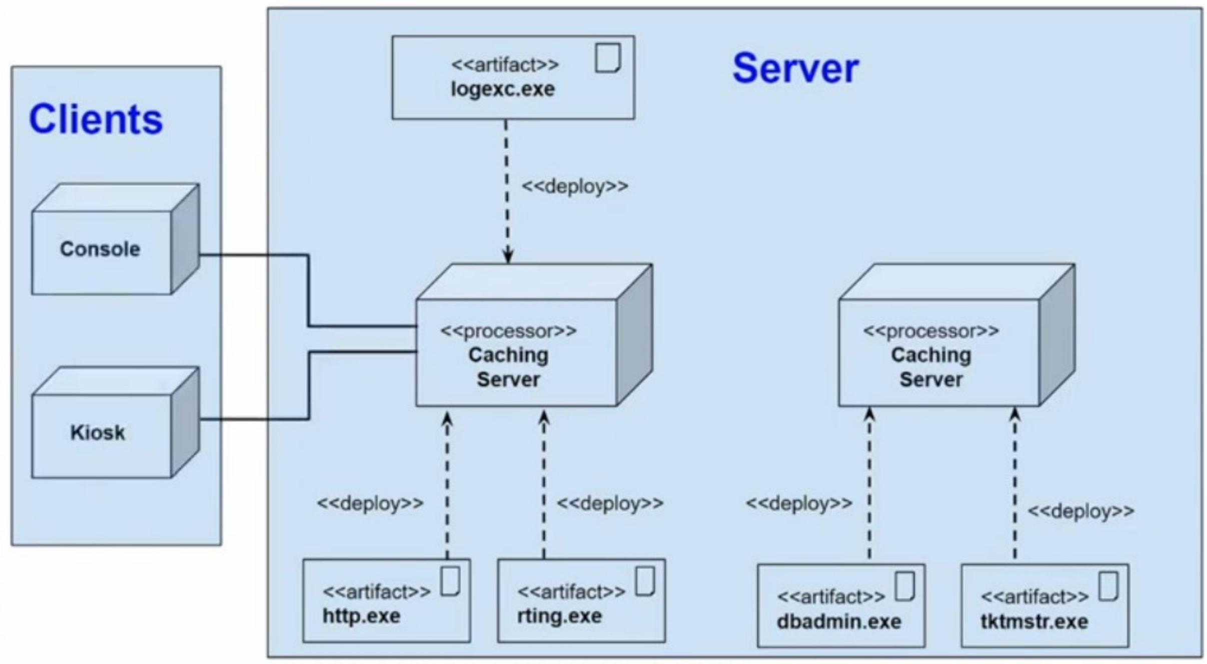


Figure 6: Architecture of any deployement diagram

To create a deployment diagram for our biometric student attendance using fingerprint we followed these steps:

1) Identify the Nodes :

- Client Devices : these are devices used by students, instructors, and admins such as smartphones, laptops, dekstops.
- Server : the server hosting the application and database.

2) Define the Components :

- Biometric System : the component handling fingerprint registration and authentication
- Attendance Management System : this is the core component handling registration, marking attendance, managing attendance, and generating reports.
- Notification System : this component is used to send notofications to users.
- User Management System : this component is for managing user accounts (students, instructors).

3) Establish the Relationships :

- this is to show how the client devices connect to the seerver.
- Also show the interaction between the server and the database server.

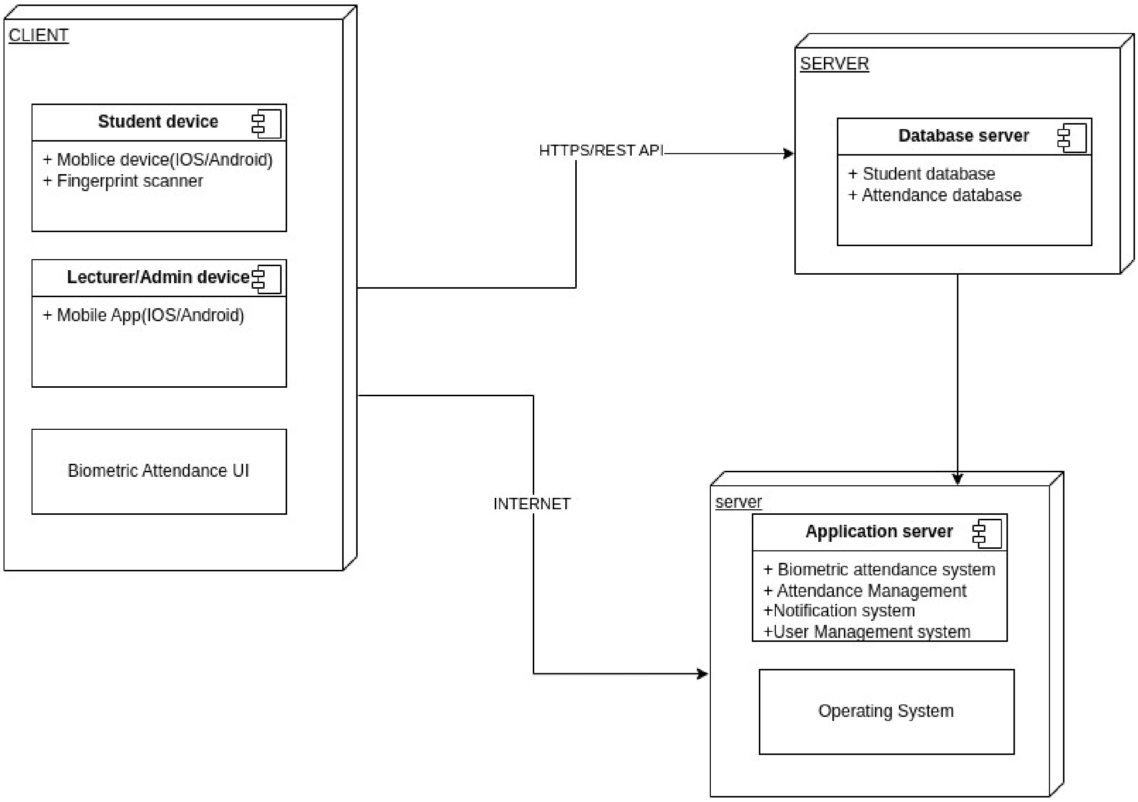


Figure 7: Deployment diagram of the AttendEase application

6. ACTIVITY DIAGRAMS

This is an important behavioral diagram used to describe the dynamic aspects of a system. It is essentially an extension version of a flow chart diagram that models the transition from one activity to another and it shows how system activities are coordinated to provide a service that can be at a different level of abstraction. *Figure 8, 9 and 10* below show us a the activities diagram of each of our stakeholders involved in the mobile application:

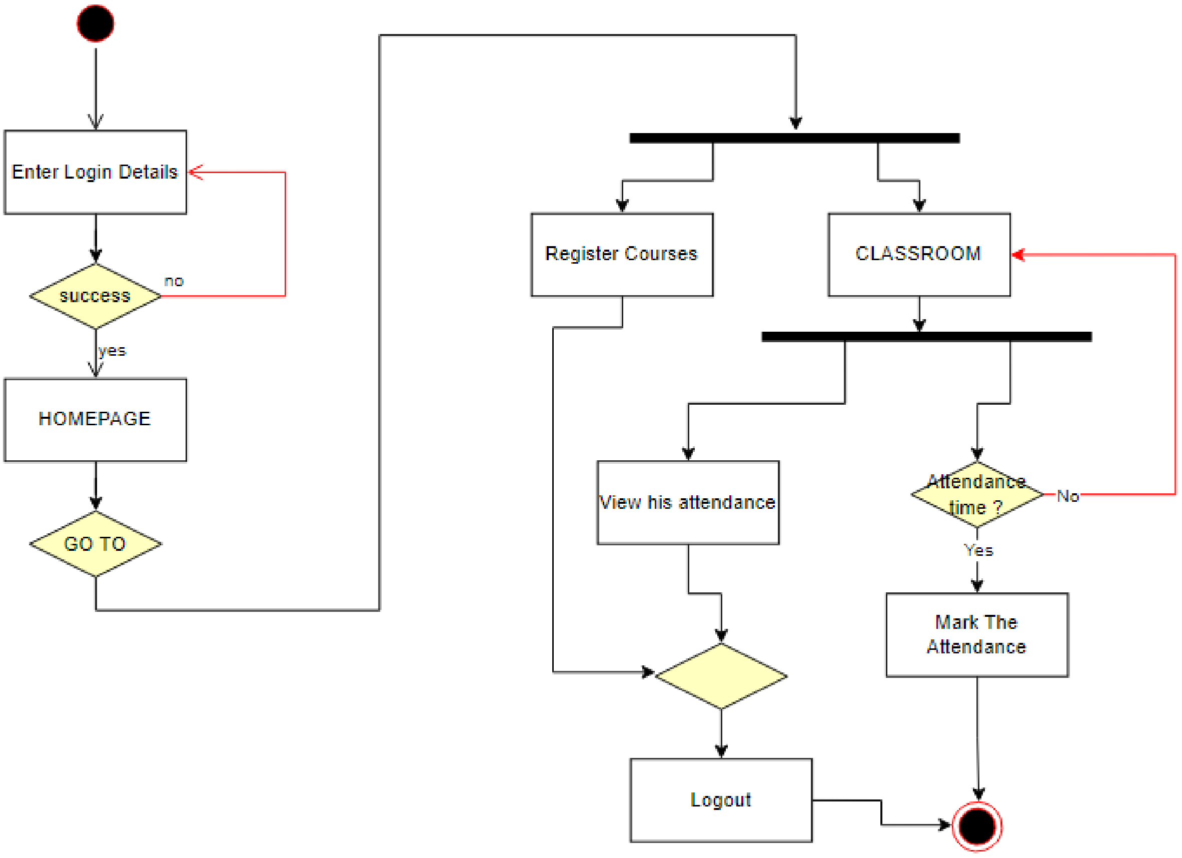


Figure 8: Activity Diagram for the student

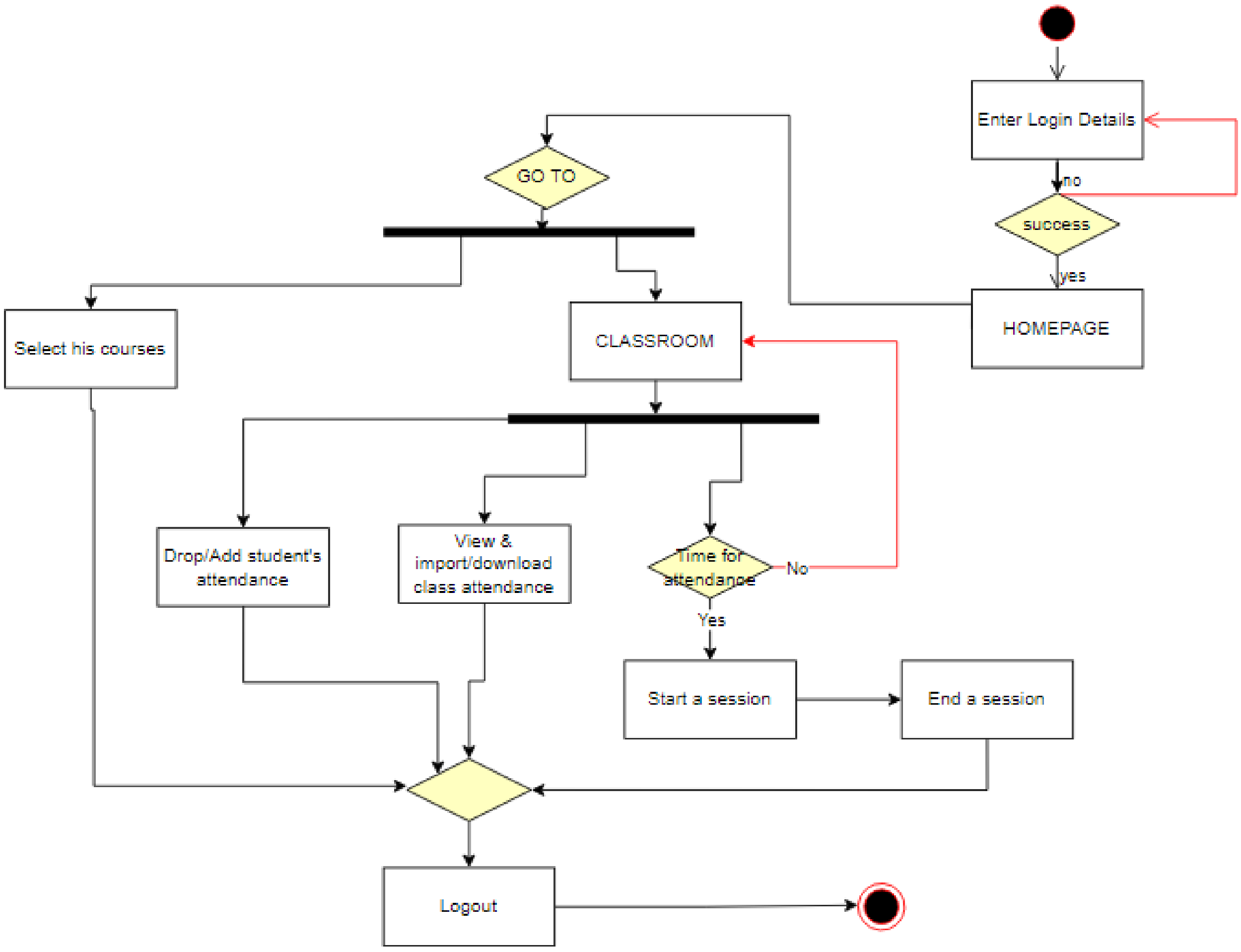


Figure 9: Activity Diagram for the Lecturer

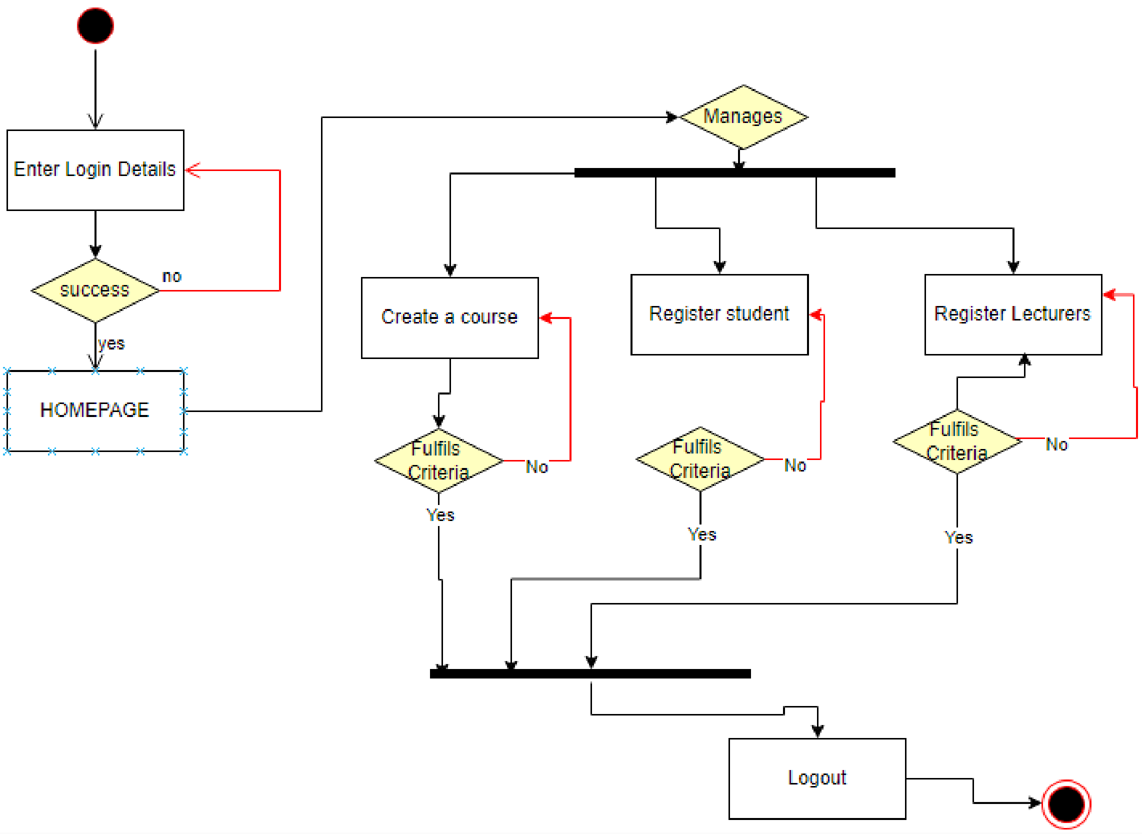


Figure 10: Activity Diagram for the administrator

CONCLUSION

The biometric student attendance mobile application outlined in this report represents a significant step forward in enhancing the efficiency and accuracy of attendance tracking within educational settings. Through the detailed system modeling and design using UML, the report has provided a holistic understanding of the application's core functionalities, user interactions, and technical implementation. The use of UML diagrams, such as use case diagrams, activity diagrams, and sequence diagrams, has enabled a clear visualization of the system's overall structure, workflow processes, and the interactions between various stakeholders and system components. By documenting these design elements, this report serves as a valuable resource for the development team, stakeholders, and future researchers interested in exploring the intricacies of biometric attendance solutions for educational institutions.

REFERENCES

- https://www.researchgate.net/publication/260317143_OBCAMS_An_Online_Biometrics-based_Class_Attendance_Management_System[1]
- https://itsourcecode.com/uml/deployment-diagram-for-student-attendance-management-system-uml/

[1] {OBCAMS}